Practical 4: Linear Programming To Investigate The Core Metabolism Of *Staphylococcus epidermidis*

Here, we will apply Linear Programming (LP) to a small model that described the core metabolism of *Staphylococcus epidermidis.* This model includes: glycolysis, TCA, ETC, and biofilm formation.

Staphylococci are:

- non-motile Gram-positive cocci,
- ubiquitous commensals of the skin and mucous membranes,
- opportunistic pathogens and the most frequent cause of infections in implanted medical devices
- Can use both oxygen and nitrate in its ETC,

We will use LP to study ATP production and biofilm formation in this organism.

The small model used here is based on a much larger genome-scale metabolic model of *Staphylococcus epidermidis (*published in Diaz Calvo *et al* (2022) Genome-Scale Metabolic Modelling Approach to Understand the Metabolism of the Opportunistic Human Pathogen Staphylococcus epidermidis RP62A . *Metabolites*, 12(2), 136. https://doi.org/10.3390/metabo12020136)

Before you start, read the documentation of the `ScrumPy` LP module (https://mudshark.brookes.ac.uk/ScrumPy/Doc/LinProg). It would be a good idea to open this page in a separate tab so that you can refer to it as you work on the practical. (Note that the examples in the documentation are **not for the same model** that you will be using; they are generic examples.)

1. Download the archive containing the model and extract the files.

   a) Start ScrumPy in the folder containing the .spy file and load the model file Sepi_biofilm.spy to create a model object.

   b) Examine how the model is created. How many reactions and metabolites are present in the model?

   c) How many media transporters are present in the model? Hints:

      - all media transporters have suffix "_mm_tx",
      - Remember the exercises from practical 2:
        - `for` statements can iterate through objects in a list (e.g. reaction names)
        - `if` statements can be used to test a given condition
        - the function *reac_name.*`endswith`*('_mm_tx')* returns true if the string *reac_name has '_mm_tx' as a suffix,*

   d) What is the name of the biofilm formation reaction? Hint: all biomass transporters have suffix "_bm_tx".

2. Set up and solve an LP problem where the objective is to minimise total flux, while producing 1 unit flux of ATP.

a) Create the LP object with the model as an argument as shown in the first steps of the documentation. Hint: lp = m.GetLP()

b) Since the optimisation direction is minimisation, and this is set by default, you don't need to change the direction, though the documentation explains how to do this.

c) Next, set the objective as total flux using the SetObjective function of the LP object. Hint: lp.SetObjective(m.sm.cnames)

d) Constrain the flux of the ATPASE-RXN to 1 using the `SetFixedFlux` function of the LP object. Hint: lp.SetFixedFlux({'ATPASE-RXN ':1.0}), where 'ATPASE-RXN ' is the name of the ATPase in the model.

e) Solve the LP. (ie lp.Solve()) The message 'Optimal solution' should appear. To obtain the solution use the LP method `GetPrimSol()`. This method returns a dictionary object of reactions in the solution as keys and flux values as values, so for convenience assign a name to this solution, e.g. sol = lp.GetPrimSol(). Hint:

- Examine the solution as follows:

```
for k in sol.keys():
    print (k, sol[k])
```

f) How many reactions are present in the solution? What is the objective value, i.e. the sum of total flux producing 1 unit flux of ATP? Hint: lp.GetObjVal()

g) What are the sources of energy/carbon in your LP solution? Hint: investigate what is being imported from the media.

h) Calculate ATP generated per unit carbon uptake.

i) Is oxygen used as terminal electron acceptor in your solution? Hint: Oxygen transporter is named as "O2_mm_tx".

3. Next we will simulate biofilm formation.

a) Generate a new LP object with the model as an argument.

b) Optimisation direction and the `SetObjective` function remains same as above problem (see 2b and 2c respectively).

c) Constrain the flux of the biomass formation reaction to -1 using the `SetFixedFlux` function of the LP object. Hint: the fixed flux is negative since biomass is **exported**.

d) Solve the LP. (ie `lp.Solve()`) The message 'Optimal solution' should appear. To obtain the solution use the LP method `lp.GetPrimSol()`. This method returns a dictionary object of reactions in the solution as keys and flux values as values, so for convenience assign a different name to this solution.

e) How many reactions are present in the solution? What is the objective value, i.e. the sum of total flux producing biofilm?

f) How many substrates are imported from the media? How many bi-products are exported along with the biofilm? Hint: all bi-product transporters have suffix “_bp_tx”.

g) Examine the change in oxygen consumption. What is the terminal electron acceptor in your solution? What might be the reason for this change in behaviour?